Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 677–680

Procedia Computer Science

The 8th International Conference on Mobile Web Information Systems (MobiWIS)

Business Process Integration using Telco Mashups

Olexiy Chudnovskyy, Hendrik Gebhardt, Frank Weinhold and Martin Gaedke

Department of Computer Science, Chemnitz University of Technology
09111 Chemnitz, Germany

Abstract

In this paper we present our approach to support business processes integration and execution using telco services. We adopt Web mashup techniques to systematically develop collaborative Web applications and utilize telco services as a well-known and ubiquitous medium for data transfer and exchange. We define a telco mashup reference architecture and derive a dedicated mashup execution platform. Finally, we present current research challenges and give an outlook of further work.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.

Keywords: Mashup; Telco services; Composition; Process Integration;

1. Introduction

A large number of novel telephony and messaging services became accessible to the Web and software developer community in the last few years [1]. These so called telco services provide unique capabilities, which can be utilized in many business process integration scenarios. Due to the ubiquity of operator networks and high availability of mobile devices data between involved parties can be transferred more efficiently and independently of location.

Currently, the integration of these functionalities into the Web applications is challenging. We identified three main problem areas: First, not all of the operator network services are exposed to Web developers. Though the number of dedicated gateways and APIs grows with every year, their heterogeneity and fast evolution complicate the development of enterprise applications [2]. Second, without adequate models and tools the integration of telephony services is a time-consuming and error-prone task [3]. And finally, the novelty of the emerging services and device capabilities requires a systematic approach and guidelines to support unskilled Web developers in the integration process [4].

In our approach we meet these challenges by applying Web mashup techniques and provide dedicated support to integrate and to consume telco services in a systematic way. Web mashups are characterized through the end-user oriented development paradigm and continuous reuse of already existing components and functionalities [5]. New goal-oriented mashups can be constructed even without programming skills leveraging the experience and building blocks produced by other developers. Thus the effort to develop telco-enabled software solutions is decreased and the quality of the solution gets improved.

The rest of the paper structures as follows: Section 2 analyses the nature of telco services. Section 3 presents our vision of telco mashups and describes the dedicated execution platform. Section 4 concludes the paper and points out current research challenges.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas. doi:10.1016/j.procs.2011.07.088

2. Understanding Telco Services

We define telco services as software services that provide communication and collaboration support. Depending on the network the services operate in, we distinguish between Internet telco services, converged services and signaling services.

Internet telco services operate exclusively in the Internet, e.g. Voice over IP (VoIP) or instant messaging. The variety of available protocols and technologies enable these services to be used in complex data transfer and workflow execution scenarios. Internet telco services provide a tool for asynchronous data transfer and synchronous voice/video communication. Furthermore, data transferred over services like instant messaging can be processed automatically and serve as a signal to initiate further execution steps.

Converged services mediate between different networks and communication protocols. A SMS message or VoIP calls from Internet to a mobile phone are examples of converged services. Converged services enable location-independent data exchange between parties, who have no access to the Internet but can communicate over other channels like operator networks. Especially processes and decision tasks, where people are involved, can benefit from capabilities of operator networks and pervasive availability of mobile devices. Monitoring and management of processes can be performed as well by notification using SMS or MMS.

Signaling services provide access to a network operator's signaling infrastructure. Examples of signaling services are notifications about incoming calls or negotiation of Quality of Service (QoS) parameters. Furthermore, signaling services can be used to establish a connection between two parties in order to initiate data transfer over alternative communication channel.

In addition, we define *device APIs* as services, which enable access to device capabilities such as cameras, microphone, location services etc. They provide additional data, which can be important or wishful for many enterprise scenarios. For example, location data from smartphones with GPS support can be utilized for decision making and task assignment process. As a result a better awareness of communication partners can be achieved. Furthermore device APIs enable mashup applications to be partially executed on the end devices and provide additional functions to the user.

In order to emphasize the potential of telco services, we analyzed representatives of the groups above and summarized their key characteristics[a]. Table 1 gives an overview over some of the application areas of telco services:

Table 1. Telco services in enterprise scenarios

Service class	Data Transferred	Synchronicity	Cost	Application Area
Internet Telco Services	Instant Messaging, Voice, Video	synchronous (Skype, Sipgate) + asynchronous (Skype IM, Jabber)	Mostly free available	Enterprise Application Integration, Business Process Integration, Business Process Monitoring and Management
Converged Services	SMS/MMS, Voice, Video	synchronous (Skype, Sipgate, Twilio, Tropo) + asynchronous (SMS services)	Mostly with costs (subscription, day passes or on message basis)	Business Process Integration, Process Execution
Signaling Services	Control Data	asynchronous (Comfone Signaling, Twilio Calls)	Mostly with costs (on minute basis)	Business Process Execution and Management
Device APIs	Data, Voice, Video	asynchronous (W3C geolocation API, W3C Messaging API)	Mostly free (or regulations within operator contract)	Business Process Execution, Business Process Monitoring

The analysis shows, that telco services provide an efficient tool for asynchronous data transfer and synchronous voice/video communication. Processes involving people can exploit the capabilities of telco services to coordinate

and execute workflows in cross-organizational scenarios. Furthermore, data transferred over operator networks can be processed by applications and be utilized for monitoring and management purposes.

In the next section we show, how telco services and device APIs can be assembled to a single software solution, so called telco mashup.

3. Telco Mashups

Telco mashups represent an enhancement of classic Web mashups and leverage the capabilities of telco services. Within a mashup, telco services are combined with other functionalities, which enable execution of both ad-hoc and complex cross-organizational workflows. We identified several layers of combination and aggregation possibilities regarding data, application logic and pieces of user interface:

- *Service Binding Layer* specifies data sources and services to be integrated into the mashup. Policies, security considerations as well as quality of service aspects have to be defined at this point to enable a cross-organisational data transfer and service invocation.
- *Data Mashup Layer* represents a step, where data coming from a number of heterogeneous sources are transformed, filtered and aggregated. The combination algorithm to be applied might be given either in form of a simple script snippet or using a dedicated mashup language, e.g. EMML [6] or DERI Pipes [7]. The data mashup enables to integrate the information coming from different organizations and departments in order to visualize workflows, execution states, relationships etc. In ad-hoc workflows the aggregated data can be used to make decisions and initiate further execution steps or processes.
- *Widget Layer* specifies graphical interface and interaction with underlying data mashups or services. The resulting components, called widgets, can be based on various standards, e.g. W3C Widgets [8], Java Portlets [9], Google Gadgets [10] etc. Different vendors and business partners can produce widgets, so that complete processes and workflows are encapsulated within one single component. To facilitate the reusability of widgets and support developers in discovery and composition tasks we store all widgets in a dedicated widget repository.
- *Workspace Layer (or UI-mashup)* is a set of inter-connected widgets with additional services and configurations regarding inter-widget communication and layout. User works with the workspace and accesses functionalities provided by added widgets. Widgets communicate with each other using a dedicated event bus and access general services implemented by telco mashup execution platform. Incoming calls or messaging services are propagated by the platform to the workspace, so each widget is notified about context changes and telco events. Inter-widget communication is a useful mechanism to transfer data between processes implemented by widgets and coordinate execution of single tasks.

Telco mashups enable collaborative work and process execution by several participants. Each mashup has its own usage policies - financial and governance rules, quality of service aspects and user profile settings. The mashup execution platform guarantees that the defined rules are adhered during the whole life cycle of the mashup. The platform is the central component within our architecture, which takes care of all aspects of mashup life cycle and facilitates the integration of telco services (Fig. 1).

The platform is distributed on the *client side* (embedded into the Web browser) and *server side*. Server side provides access to embedded telco services and mashup management facilities. Upon request, new mashups are instantiated based on their configuration (stored in *mashup repository*) and *user profile settings* (security, billing and service level agreements). The execution of mashups is managed by the *life cycle manager* component, which guarantees, that charging and QoS settings, predefined availability as well as security and federation rules are respected. The *communication manager* hosted on the server side of the execution platform provides endpoints for mobile devices, manages incoming calls and routes them to corresponding mashup instances. It enables signaling of process statuses and data transfer using mobile devices and pervasive operator services (SMS/MMS/Voice call etc). Incoming events are mapped to the corresponding mashup instance and can be consumed by the widgets assembled within the workspace. Contrariwise widgets use telco services integrated in the platform to communicate with agents available only in operator networks.

As such, the execution platform provides facilities to operate and to manage telco-enabled mashups. It transcends technological communication constrains and enables integration of business processes in a new seamless way.

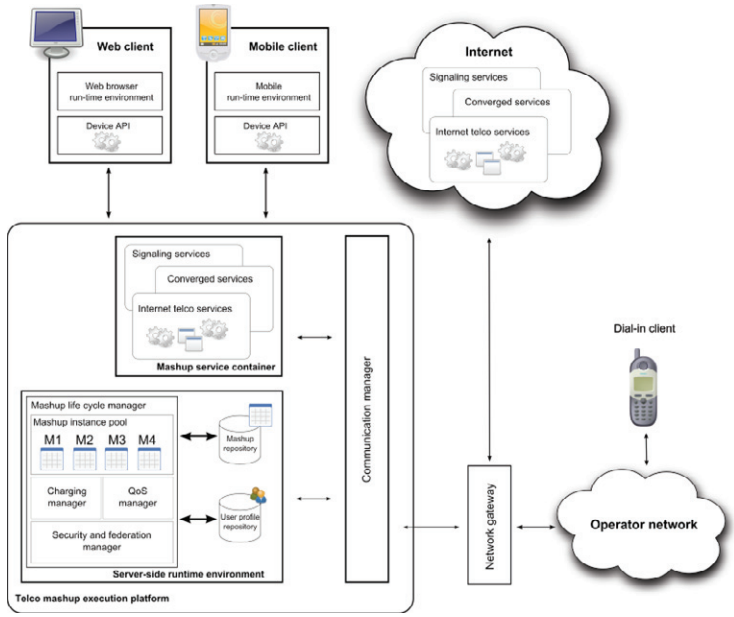

Fig. 1. Telco mashup reference architecture

4. Conclusions and Outlook

In this paper we have presented telco mashups and their usage in business process integration scenarios. We showed our current progress and findings as well as initial telco mashup reference architecture. We discussed the impact of different mashup layers on business process integration and described the execution platform, which enables integration of telco-services in a seamless way. Currently we are working on specification of telco mashup description language, which should cover all the aspects of mashup development process and its lifecycle. Furthermore we are going to implement several prototypes of telco mashups with support of some simple essential telco services.

Acknowledgements

This work was supported by funds from the European Commission (project OMELETTE, contract no. 257635)

References

1. ProgrammableWeb - Mashups, APIs, and the Web as Platform, http://www.programmableweb.com/, (2011-06-09).
2. Daniel, F., Matera, M. Turning Web Applications into Mashup Components: Issues, Models, and Solutions. Springer Berlin Heidelberg, Berlin, Heidelberg (2009).
3. Sienel, J., Martín, A.L., Zorita, C.B., Martínez, B.C. OPUCE : A Telco-Driven Service Mash-Up Approach. Bell Labs Technical Journal. 14, pp. 203-218 (2009).
4. Cappiello, C., Daniel, F., Matera, M., Picozzi, M., Weiss, M. Enabling End User Development through Mashups : Requirements , Abstractions and Innovation Toolkits. Development. pp. 1-16.
5. Yu, J., Benatallah, B., Casati, F., Daniel, F. Understanding Mashup Development. IEEE Internet Computing. 12, pp. 44-52 (2008).
6. Viswanathan, A. Mashups and the Enterprise Mashup Markup Language (EMML), http://www.drdobbs.com/article/printableArticle.jhtml?articleId=224300049&dept_url=/java/, (2010-10-18).
7. Phuoc, D.L., Polleres, A., Tummarello, G., Morbidoni, C. DERI Pipes: visual tool for wiring Web data sources. (2008).
8. Widget Packaging and Configuration, http://www.w3.org/TR/widgets/, (2011-06-09).
9. Sun Microsystems: Introduction to JSR 168—The Java Portlet Specification, http://developers.sun.com/portalserver/reference/techart/jsr168/, (2011-06-09).
10. Gadgets Specification - Gadgets API - Google Code, http://code.google.com/intl/de-DE/apis/gadgets/docs/spec.html, (2011-06-09).